AMENDMENTS TO THE CLAIMS

1. (Currently Amended) A computer-implemented method for testing a parameterizable logic core, comprising:

randomly generating a set of parameter values for the logic core;

generating a netlist from the set of parameter values and logic core;

simulating circuit behavior with the netlist;

cloning the set of parameter values;

mutating the set of parameter values, whereby a mutated set of parameter values is produced;

generating a new netlist from the mutated set of parameter values and logic core; and

simulating circuit behavior with the new netlist.

2. (Original) The method of claim 1, further comprising for each parameter:

generating a random parameter value within predetermined upper and lower limits associated with the parameter; and

generating a new random parameter value if the random parameter value fails to meet predetermined criteria.

3. (Original) The method of claim 2, further comprising:

assigning respective probabilities to one or more numbers between the upper and lower limits for one or more of the parameters; and

generating the random parameter value as a function of the probabilities.

4. (Original) The method of claim 3, further comprising:

providing the parameter value as input to a graphical user interface;

generating random replacement values for invalid parameter values detected by the graphical user interface.

5. (Original) The method of claim 1, further comprising:

providing the set of parameter values to a graphical user interface; and

identifying invalid parameter values with the graphical user interface.

6. (Original) The method of claim 5, further comprising:

generating random replacement parameter values for the invalid parameters; and

repeating the steps of providing the replacement values to the graphical user interface, identifying invalid parameter values, and generating random replacements until all the parameter values are valid.

7. (Original) The method of claim 5, further comprising:

selecting a random order in which to provide parameter values to the graphical user interface; and

providing the parameters one-by-one as input to the graphical user interface.

Claim 8-10. (Cancelled)

11. (Currently Amended) The method of claim 1 [[10]], wherein the set of parameter values is cloned and mutated only if an error is detected in simulating the circuit behavior.

12. (Original) The method of claim 11, further comprising repetitively cloning and mutating sets of parameters when errors are detected in simulating the circuit behavior, whereby multiple generations of sets of parameters are created.

13. (Original) The method of claim 12, wherein one or more of the parameter values in a parameter set are mutated.

14. (Original) The method of claim 13, wherein a number of parameter values mutated in a set of parameters is a function of a number of generations previously created.

15. (Original) The method of claim 1, further comprising:

identifying which parameters of a graphical user interface are editable; and

providing the set of parameter values to the graphical user interface.

16. (Original) The method of claim 15, further comprising:

selecting a random order in which to provide the parameter values to the graphical user interface; and

providing the values one-by-one as input to the graphical user interface.

17. (Original) The method of claim 1, further comprising:

accumulating respective numbers of tests having been performed using different parameter values

accumulating respective numbers of tests failed using each of the parameter values; and

highlighting parameters having numbers of failed tests equal to the number of tests.

18. (Currently Amended) A system for testing a parameterizable logic core, comprising:

a test controller configured and arranged to randomly generate a set of parameter values for the logic core;

a core generator coupled to the test controller, the core generator configured and arranged to generate a netlist from the logic core and set of parameter values;

a simulator coupled to the test controller, the simulator configured and arranged to simulating circuit behavior with the netlist and a predetermined test bench;

wherein the test controller is further configured to clone the set of parameter values and mutate the set of parameter values, whereby a mutated set of parameter values is produced, the core generator is configured to generate a new netlist from the mutated set of parameter values and logic core, and the simulator is configured to simulate circuit behavior with the new netlist.

19. (Currently Amended) A system for testing a parameterizable logic core, comprising:

a test controller including a GUI-driver, the GUI-driver configured and arranged to randomly generate a set of parameter values for the logic core;

a core generator including a GUI coupled to the GUI-driver, the core generator configured and arranged to generate a netlist from the logic core and set of parameter values;

a simulator coupled to the test controller, the simulator configured and arranged to simulating circuit behavior with the netlist and a predetermined test bench;

wherein the GUI-driver is further configured to clone the set of parameter values and mutate the set of parameter values, whereby a mutated set of parameter values is produced, the core generator is configured to generate a new netlist from the mutated set of parameter values and logic core, and the simulator is configured to simulate circuit behavior with the new netlist.

20. (Currently Amended) An apparatus for testing a parameterizable logic core, comprising:

means for randomly generating a set of parameter values for the logic core;

means for generating a netlist from the set of parameter values and logic core; and

means for simulating circuit behavior with the netlist[[.]];

means for cloning the set of parameter values;

means for mutating the set of parameter values, whereby a mutated set of parameter values is produced;

means for generating a new netlist from the mutated set of parameter values and logic core; and

means for simulating circuit behavior with the new netlist.